

**SYSTEM AND METHOD FOR ON-BOARD TIMING  
MARGIN TESTING OF MEMORY MODULES**

**TECHNICAL FIELD**

[001] The present invention relates to a processor-based system, and more particularly, to a processor-based system having a memory module with a memory hub coupling several memory devices to a processor or other memory access devices.

**BACKGROUND OF THE INVENTION**

[002] Processor-based systems, such as computer systems, use memory devices, such as dynamic random access memory ("DRAM") devices, to store instructions and data that are accessed by a processor. These memory devices are typically used as system memory in a computer system. In a typical computer system, the processor communicates with the system memory through a processor bus and a memory controller. The processor issues a memory request, which includes a memory command, such as a read command, and an address designating the location from which data or instructions are to be read. The memory controller uses the command and address to generate appropriate command signals as well as row and column addresses, which are applied to the system memory. In response to the commands and addresses, data is transferred between the system memory and the processor. The memory controller is often part of a system controller, which also includes bus bridge circuitry for coupling the processor bus to an expansion bus, such as a PCI bus.

[003] Although the operating speed of memory devices has continuously increased, this increase in operating speed has not kept pace with increases in the operating speed of processors. Even slower has been the increase in operating speed of memory controllers coupling processors to memory devices. The relatively slow speed of memory controllers and memory devices limits the data bandwidth between the processor and the memory devices.

[004] In addition to the limited bandwidth between processors and memory devices, the performance of computer systems is also limited by latency problems that increase the time required to read data from system memory devices. More specifically, when a

memory device read command is coupled to a system memory device, such as a synchronous DRAM (“SDRAM”) device, the read data are output from the SDRAM device only after a delay of several clock periods. Therefore, although SDRAM devices can synchronously output burst data at a high data rate, the delay in initially providing the data can significantly slow the operating speed of a computer system using such SDRAM devices.

[005] One approach to alleviating the memory latency problem is to use multiple memory devices coupled to the processor through a memory hub. In a memory hub architecture, a system controller or memory hub controller is coupled to several memory modules, each of which includes a memory hub coupled to several memory devices. The memory hub efficiently routes memory requests and responses between the controller and the memory devices. Computer systems employing this architecture can have a higher bandwidth because a processor can access one memory module while another memory module is responding to a prior memory access. For example, the processor can output write data to one of the memory modules in the system while another memory module in the system is preparing to provide read data to the processor. The operating efficiency of computer systems using a memory hub architecture can make it more practical to vastly increase data bandwidth of a memory system. A memory hub architecture can also provide greatly increased memory capacity in computer systems.

[006] Although there are advantages to utilizing a memory hub for accessing memory devices, the design of the hub memory system, and more generally, computer systems including such a memory hub architecture, becomes increasingly difficult. For example, in many hub based memory systems, the processor is coupled to the memory via a high speed bus or link over which signals, such as command, address, or data signals, are transferred at a very high rate. However, as transfer rates increase, the time for which a signal represents valid information is decreasing. As commonly referenced by those ordinarily skilled in the art, the window or “eye” for when the signals are valid decreases at higher transfer rates. With specific reference to data signals, the “data eye” decreases. As understood by one skilled in the art, the data eye for each of the data signals defines the actual duration that

each signal is valid after various factors affecting the signal are considered, such as timing skew, voltage and current drive capability, and the like. In the case of timing skew of signals, it often arises from a variety of timing errors such as loading on the lines of the bus and the physical lengths of such lines.

[007] As data eyes of the signals decrease at higher transfer rates, it is possible that one or more of a group of signals provided in parallel will have arrival times such that not all signals are simultaneously valid at a receiving entity, and thus cannot be successfully captured that entity. For example, where a plurality of signals are provided in parallel over a bus, the data eye of one or more of the particular signals do not overlap with the data eyes of the other signals. In this situation, the signals having non-overlapping data eyes are not valid at the same time as the rest of the signals, and consequently, cannot be successfully captured by the receiving entity.

[008] Clearly, as those ordinarily skilled in the art will recognize, the previously described situation is unacceptable. As it is further recognized by those familiar in the art of high speed digital systems, signal timing is an issue that has become increasingly more significant in the design of systems capable of transferring and transmitting information at high speeds because signal timing can be affected by many things. As a result, testing memory devices that are subject to variations in the timing relationships between signals, whether command, address, or data signals, is becoming a more critical step in the production of memory devices that are to be used in a high-speed digital system.

[009] Production testing is typically accomplished using sophisticated testing equipment that is coupled to several memory devices to simultaneously test the devices. During testing, the testing equipment couples signals to and from integrated circuit nodes sometimes using a test probe that makes electrical contact with the circuit nodes. However, such probes can introduce loading effects that change the characteristic of the signals being evaluated. Although probes are specifically designed to have high impedance and low capacitance to minimize loading issues and the introduction of noise, there is still in many cases, an unacceptable level of loading that changes the character of a signal to such a degree that it cannot be accurately evaluated. Also, the propagation delays in coupling

signals between the integrated circuit nodes and the test equipment may introduce delays that make it impossible to accurately determine timing relationships within the integrated circuit. For example, the testing equipment may apply memory command and address signals to the memory device with a specific range of timing relationships to write data signals that are also applied to the memory device. If the memory device is able to properly read the write data, the assumption is made that the memory device is able to operate within this range of timing relationships. However, there can be no assurance that the command and address signals and the write data signals are actually coupled to circuit nodes in the memory device with this same range of timing relationships. Therefore, the memory device may not actually function properly with the timing relationships that were used during the testing.

[010] The difficulty in accurately controlling and/or determining the actual timing relationships between signals applied to or internal to memory devices is exacerbated when the memory devices are accessible only through interface circuitry. For example, where several memory devices are coupled to a memory hub, the memory devices are accessible only through the memory hub. Although production testing equipment may be able to accurately control and determine the timing relationships between signals applied to the interface circuitry, such equipment cannot control or determine the timing relationships of the signals in the memory devices after the devices have been packaged with the interface circuitry so that the signals coupled to and from the memory devices must be coupled through the interface circuitry.

[011] Another problem that can be encountered in testing high-speed memory devices using conventional production testing equipment is associated with obtaining control over the memory bus in order to perform evaluation. Again, this problem is exacerbated when memory devices are accessible to production testing equipment only through interface circuitry, such as a memory hub. The ability to evaluate a memory system often requires that specific signals of interest can be captured and analyzed by obtaining control of the memory bus and monitoring the interaction of the signal with the bus itself. Unless control over the memory bus can be obtained, analysis becomes a difficult task. However,

obtaining control over the memory bus is a difficult task in itself because conventional approaches often interfere with the normal operation of the computer system, thus, preventing accurate analysis of the memory system under true, normal operating conditions.

[012] There is therefore a need for a system and method that allows memory devices to be tested through interface circuitry in a manner that does not affect the normal operation of the memory device and that provides accurate control and determination of timing relationships between signals applied to or received from the memory device. In particular, there is a need for a system and method that allows the timing relationships between signals applied to or received from memory devices to be controlled and determined even though the memory devices are contained in a memory hub module.

#### SUMMARY OF THE INVENTION

[013] A memory module for a processor-based system includes a plurality of memory devices coupled to a memory hub. The memory hub includes a link interface for receiving memory requests for access to the memory devices and at least one memory device interface coupled to the memory devices. The memory device interface couples write memory requests and write data to the memory devices, and couples read memory requests to the memory device and read data from the memory device. The memory hub also includes a self-test module coupled to the memory devices. The self-test module is operable to couple to the memory devices a series of corresponding first and second signals. Significantly, the self-test module alters the relative timing between when some of the corresponding first and second signals in the series are coupled to the memory devices. The self-test module also receives output signals from the memory devices and determines based on the received output signals whether the memory device properly responded to the series of first and second signals. The self-test module may also vary the time when the output signals are registered in the memory hub relative to a strobe signal received with the output signals. The operation of the memory devices may be synchronized to a clock signal generated by the memory hub. The frequency of the clock signal may be varied to verify the operation of the memory devices at an abnormally high operating speed.

### BRIEF DESCRIPTION OF THE DRAWINGS

- [014] Figure 1 is a block diagram of a computer system that includes several memory modules having a memory hub architecture according to various embodiments of the present invention.
- [015] Figure 2 is a block diagram of a memory hub according to an embodiment of the present invention that may be used with the memory modules in the computer system of Figure 1 or in some other system.
- [016] Figure 3 is a block diagram of one embodiment of a self test module used in the memory hub of Figure 2.

### DETAILED DESCRIPTION OF THE INVENTION

- [017] Embodiments of the present invention are directed to a memory hub module having the capability to self-test signal timing relationships. Certain details are set forth below to provide a sufficient understanding of various embodiments of the invention. However, it will be clear to one skilled in the art that the invention may be practiced without these particular details. In other instances, well-known circuits, control signals, and timing protocols have not been shown in detail in order to avoid unnecessarily obscuring the invention.
- [018] A computer system 100 according to one embodiment of the invention is shown in Figure 1. The computer system 100 includes a processor 104 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 includes a processor bus 106 that normally includes an address bus, a control bus, and a data bus. The processor bus 106 is typically coupled to cache memory 108, which, is typically static random access memory ("SRAM"). Finally, the processor bus 106 is coupled to a system controller 110, which is also sometimes referred to as a bus bridge.
- [019] The system controller 110 serves as a communications path to the processor 104 for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller 112, which is, in turn,

coupled to a video terminal 114. The system controller 110 is also coupled to one or more input devices 118, such as a keyboard or a mouse, to allow an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or retrieve data from internal or external storage media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

[020] The system controller 110 includes a memory hub controller 128 that is coupled to several memory modules 130a,b...n, which serve as system memory for the computer system 100. The memory modules 130 are preferably coupled to the memory hub controller 128 through a high-speed link 134, which may be an optical or electrical communication path or some other type of communications path. In the event the high-speed link 134 is implemented as an optical communication path, the optical communication path may be in the form of one or more optical fibers. In such case, the memory hub controller 128 and the memory modules will include an optical input/output port or separate input and output ports coupled to the optical communication path. The memory modules 130 are shown coupled to the memory hub controller 128 in a multi-drop arrangement in which the single high-speed link 134 is coupled to all of the memory modules 130. However, it will be understood that other topologies may also be used. For example, a point-to-point coupling arrangement may be used in which a separate high-speed link (not shown) is used to couple each of the memory modules 130 to the memory hub controller 128. A switching topology may also be used in which the memory hub controller 128 is selectively coupled to each of the memory modules 130 through a switch (not shown). Other topologies that may be used will be apparent to one skilled in the art.

[021] Each of the memory modules 130 includes a memory hub 140 for controlling access to eight memory devices 148, which, in the example illustrated in Figure 1, are synchronous dynamic random access memory ("SDRAM") devices. However, a fewer or

greater number of memory devices 148 may be used, and memory devices other than SDRAM devices may also be used. The memory hub 140 is coupled to each of the system memory devices 148 through a bus system 150, which normally includes a control bus, an address bus, and a data bus. However, other bus systems, such as a bus system using a shared command/address bus, may also be used

[022] Figure 2 shows an embodiment of a memory hub 200 according to the present invention, which can be used as the memory hub 140 of Figure 1. The memory hub 200 is being coupled to four memory devices 240a-d, which, in the present example are conventional SDRAM devices. In an alternative embodiment, the memory hub 200 is coupled to four different banks of memory devices, rather than merely four different memory devices 240a-d, with each bank typically having a plurality of memory devices. However, for the purpose of providing an example, the present description will be with reference to the memory hub 200 coupled to the four memory devices 240a-d. It will be appreciated that the necessary modifications to the memory hub 200 to accommodate multiple banks of memory is within the knowledge of those ordinarily skilled in the art.

[023] Further included in the memory hub 200 are link interfaces 210a-d, which may be used to couple the memory hub 200 to respective processors or other memory access devices. In the embodiment shown in Figure 1, only one memory access device, and hence only one link interface 210a, is used. The memory hub 200 also includes link interfaces 212a-d for coupling the memory module on which the memory hub 200 is located to other memory modules (not shown). These link interfaces 212a-d are not used in the embodiment of Figure 1. In any case, the link interfaces 210a-d and 212a-d are preferably coupled to a first high speed data link 220 and a second high speed data link 222, respectively. As previously discussed with respect to Figure 1, the high speed data links 220, 222 can be implemented using an optical or electrical communication path or some other type of communication path. The link interfaces 210a-d, 212a-d are conventional, and include circuitry used for transferring data, command, and address information to and from the high speed data links 220, 222. As well known, such circuitry includes transmitter and receiver logic known in the art. It will be appreciated that those ordinarily skilled in the



art have sufficient understanding to modify the link interfaces 210a-d, 212a-d to be used with specific types of communication paths, and that such modifications to the link interfaces 210a-d, 212a-d can be made without departing from the scope of the present invention. For example, in the event the high-speed data link 220, 222 is implemented using an optical communications path, the link interfaces 210a-d, 212a-d will include an optical input/output port that can convert optical signals coupled through the optical communications path into electrical signals.

[024] The link interfaces 210a-d, 212a-d are coupled to a switch 260 through a plurality of bus and signal lines, represented by busses 214. The busses 214 are conventional, and include a write data bus and a read data bus, although a single bi-directional data bus may alternatively be provided to couple data in both directions through the link interfaces 210a-d, 212a-d. It will be appreciated by those ordinarily skilled in the art that the busses 214 are provided by way of example, and that the busses 214 may include fewer or greater signal lines, such as further including a request line and a snoop line, which can be used for maintaining cache coherency.

[025] The link interfaces 210a-d, 212a-d include circuitry that allow the memory hub 140 to be connected in the system memory in a variety of configurations. For example, the multi-drop arrangement, as shown in Figure 1, can be implemented by coupling each memory module to the memory hub controller 128 through either the link interfaces 210a-d or 212a-d. Alternatively, a point-to-point or daisy chain configuration can be implemented by coupling the memory modules in series. For example, the link interfaces 210a-d can be used to couple a first memory module and the link interfaces 212a-d can be used to couple a second memory module. The memory module coupled to a processor, or system controller, will be coupled thereto through one set of the link interfaces and further coupled to another memory module through the other set of link interfaces. In one embodiment of the present invention, the memory hub 200 of a memory module is coupled to the processor in a point-to-point arrangement in which there are no other devices coupled to the connection between the processor 104 and the memory hub 200. This type of interconnection provides better signal coupling between the processor 104 and the memory hub 200 for several reasons,

including relatively low capacitance, relatively few line discontinuities to reflect signals and relatively short signal paths.

[026] The switch 260 is further coupled to four memory interfaces 270a-d which are, in turn, coupled to the memory devices 240a-d, respectively. By providing a separate and independent memory interface 270a-d for each memory device 240a-d, respectively, the memory hub 200 avoids bus or memory bank conflicts that typically occur with single channel memory architectures. The switch 260 is coupled to each memory interface through a plurality of bus and signal lines, represented by busses 274. The busses 274 include a write data bus, a read data bus, and a request line. However, it will be understood that a single bi-directional data bus may alternatively be used instead of a separate write data bus and read data bus. Moreover, the busses 274 can include a greater or lesser number of signal lines than those previously described.

[027] In an embodiment of the present invention, each memory interface 270a-d is specially adapted to the memory devices 240a-d to which it is coupled. More specifically, each memory interface 270a-d is specially adapted to provide and receive the specific signals received and generated, respectively, by the memory device 240a-d to which it is coupled. Also, the memory interfaces 270a-d are capable of operating with memory devices 240a-d operating at different clock frequencies. As a result, the memory interfaces 270a-d isolate the processor 104 from changes that may occur at the interface between the memory hub 230 and memory devices 240a-d coupled to the memory hub 200, and it provides a more controlled environment to which the memory devices 240a-d may interface.

[028] The switch 260 coupling the link interfaces 210a-d, 212a-d and the memory interfaces 270a-d can be any of a variety of conventional or hereinafter developed switches. For example, the switch 260 may be a cross-bar switch that can simultaneously couple link interfaces 210a-d, 212a-d and the memory interfaces 270a-d to each other in a variety of arrangements. The switch 260 can also be a set of multiplexers that do not provide the same level of connectivity as a cross-bar switch but nevertheless can couple the some or all of the link interfaces 210a-d, 212a-d to each of the memory interfaces 270a-d. The

switch 260 may also includes arbitration logic (not shown) to determine which memory accesses should receive priority over other memory accesses. Bus arbitration performing this function is well known to one skilled in the art.

[029] With further reference to Figure 2, each of the memory interfaces 270a-d includes a respective memory controller 280, a respective write buffer 282, and a respective cache memory unit 284. The memory controller 280 performs the same functions as a conventional memory controller by providing control, address and data signals to the memory device 240a-d to which it is coupled and receiving data signals from the memory device 240a-d to which it is coupled. However, the nature of the signals sent and received by the memory controller 280 will correspond to the nature of the signals that the memory devices 240a-d are adapted to send and receive. The cache memory unit 284 includes the normal components of a cache memory, including a tag memory, a data memory, a comparator, and the like, as is well known in the art. The memory devices used in the write buffer 282 and the cache memory unit 284 may be either DRAM devices, static random access memory (“SRAM”) devices, other types of memory devices, or a combination of all three. Furthermore, any or all of these memory devices as well as the other components used in the cache memory unit 284 may be either embedded or stand-alone devices.

[030] The write buffer 282 in each memory interface 270a-d is used to store write requests while a read request is being serviced. In such a system, the processor 104 can issue a write request to a system memory device 240a-d even if the memory device to which the write request is directed is busy servicing a prior write or read request. The write buffer 282 preferably accumulates several write requests received from the switch 260, which may be interspersed with read requests, and subsequently applies them to each of the memory devices 240a-d in sequence without any intervening read requests. By pipelining the write requests in this manner, they can be more efficiently processed since delays inherent in read/write turnarounds are avoided. The ability to buffer write requests to allow a read request to be serviced can also greatly reduce memory read latency since read requests can be given first priority regardless of their chronological order.

[031] The use of the cache memory unit 284 in each memory interface 270a-d allows the processor 104 to receive data responsive to a read command directed to a respective system memory device 240a-d without waiting for the memory device 240a-d to provide such data in the event that the data was recently read from or written to that memory device 240a-d. The cache memory unit 284 thus reduces the read latency of the system memory devices 240a-d to maximize the memory bandwidth of the computer system. Similarly, the processor 104 can store write data in the cache memory unit 284 and then perform other functions while the memory controller 280 in the same memory interface 270a-d transfers the write data from the cache memory unit 284 to the system memory device 240a-d to which it is coupled.

[032] Further included in the memory hub 200 is a self-test module 290 coupled to the switch 260 through a test bus 292. The self-test module 290 is further coupled to a maintenance bus 296, such as a System Management Bus (SMBus) or a maintenance bus according to the Joint Test Action Group (JTAG) and IEEE 1149.1 standards. Both the SMBus and JTAG standards are well known by those ordinarily skilled in the art. Generally, the maintenance bus 296 provides a user access to the self-test module 290 in order to set memory testing parameters and receive test results. For example, the user can couple a separate PC host via the maintenance bus 296 to set the relative timing between signals that are applied to the memory devices 240. Similarly, data indicative of the relative timing between signals that are received from the memory devices 240 can be coupled to the PC host via the maintenance bus 296. By using the maintenance bus 296 to set test parameters and output test results, issues related to the use of external testing equipment and test probes, as previously discussed, can be avoided. It will be appreciated that the maintenance bus 296 can be modified from conventional bus standards without departing from the scope of the present invention. It will be further appreciated that the self-test module 290 should accommodate the standards of the maintenance bus 296, where such a standard maintenance bus is employed. For example, the self-test module 296 should have a maintenance bus interface that is compliant with the JTAG bus standard where such a maintenance bus 292 is used.

[033] Further included in the memory hub 200 may be a DMA engine 286 coupled to the switch 260 through a bus 288. The DMA engine 286 enables the memory hub 200 to move blocks of data from one location in the system memory to another location in the system memory without intervention from the processor 104. The bus 288 includes a plurality of conventional bus lines and signal lines, such as address, control, data busses, and the like, for handling data transfers in the system memory. Conventional DMA operations well known by those ordinarily skilled in the art can be implemented by the DMA engine 286. A more detailed description of a suitable DMA engine can be found in commonly assigned, co-pending U.S. Patent Application No. \_\_\_\_\_, entitled APPARATUS AND METHOD FOR DIRECT MEMORY ACCESS IN A HUB-BASED MEMORY SYSTEM, filed on \_\_\_\_\_, which is incorporated herein by reference. As described in more detail in the aforementioned patent application, the DMA engine 286 is able to read a link list in the system memory to execute the DMA memory operations without processor intervention, thus, freeing the processor 104 and the bandwidth limited system bus from executing the memory operations. The DMA engine 286 can also include circuitry to accommodate DMA operations on multiple channels, for example, for each of the system memory devices 240a-d. Such multiple channel DMA engines are well known in the art and can be implemented using conventional technologies.

[034] The self-test module 290 and the DMA engine 286 are preferably embedded circuits in the memory hub 200. However, including a separate self-test module 290 and a separate DMA engine 286 coupled to the memory hub 200 is also within the scope of the present invention.

[035] Embodiments of the present invention provide an environment for varying the timing and timing relationships of signals applied to the memory devices 240, as well as determining the timing and timing relationships between signals received from the memory devices 240. For example, the self-test module 290 may cause a clock signal having a variable frequency to be applied to the memory devices 240 in the event, for example, that the memory devices 240 are synchronous memory devices, such as SDRAMs. By ensuring during production testing that the memory devices 240 can operate properly at an

abnormally high frequency, the memory devices 240 can be counted on to operate at a lower clock frequency during normal operation. As another example, the self-test module 290 can cause a data signals to be abnormally skewed relative to a data strobe signal during production testing. If the memory device 240 operates properly with this timing skew, it can reasonably be assured of operating properly in the presence of normal signal skews. Varying the timing or relative timing of signals also allows for “speed grading” of memory modules during production testing. Other variations in the timing of signals or the relative timing of signals for test purposes will be apparent to one skilled in the art.

[036] Figure 3 illustrates a self-test module 300 according to an embodiment of the present invention that can be used as the self-test module 296 shown in Figure 2. It will be appreciated that Figure 3 is a functional block diagram representative of a suitable self-test module and is not intended to limit the scope of the present invention. The functional blocks shown in Figure 3 are conventional, and can be implemented using well known techniques and circuitry. It will be further appreciated that control signals and other functional blocks have been omitted from Figure 3 in order to avoid unnecessarily obscuring the present invention, and that the description provided herein is sufficient to enable those ordinarily skilled in the art to practice the invention.

[037] Included in the self-test module 300 is a memory sequencer 304 that generates properly timed signals for controlling the operation of the memory devices 240 (Figure 2). However, in alternative embodiments, the DMA engine 286 may be used for this purpose. The nature of the signals generated by the memory sequencer 304 will, of course, be determined by the nature of the signals used by the memory devices 240. The self-test module 300 is operated primarily by a BIST engine 310, which may be a logic circuit, such as a state machine, a simple processor executing test code, or some other type of device. As will be explained, the BIST engine 310 determines the timing relationship between signals applied to the memory devices 240.

[038] One output of BIST engine 310 is applied to a pattern generator 314 of conventional design that generates signals corresponding to a pattern of data that will be written to one of the memory devices 240. The data signals from the pattern generator 314

are applied to flip-flops 318 (represented by the single flip-flop 318 shown in Figure 3), which is clocked by an internal CLK signal (“ICLK”) generated at the output of a write delay line 320. The delay line 240, in turn, receives a CLK signal produced by a clock generator 324. The CLK signal is delayed to produce the ICLK signal by a magnitude determined by a delay control signal from the BIST engine 310. The frequency of the CLK signal produced by the clock generator 324 preferably varies and is determined by a clock control signal from the BIST engine 310. Thus, data signals are output from the flip-flops 318 at a variable rate, as determined by the clock control signal, and/or at a variable timing, as determined by the delay control signal. The data signals from the flip-flops 318 are applied to data input terminals of the memory devices 240 along with the CLK signal, which may serve as a write data strobe signal.

[039] Although not specifically shown in Figure 3, the self-test module 300 can also vary the timing and rate at which control and address signals are applied to the memory devices 240 along with the CLK signal or other timing signal.

[040] Read data from the memory device 240 is applied to the data input of a flip-flop 330, which also receives a delayed read data strobe signal at the output of a read delay line 334. The delay line 334 receives a read data strobe signal from the memory device 240. The magnitude of the delay provided by the read delay line 334 is determined by a read delay control signal coupled from the BIST engine 310. The flip-flop 330 then applies the read data to a compare circuit 340. The compare circuit 340 receives not only the read data from the memory device 240, but also receives the pattern of data from the pattern generator 314. The compare circuit 340 compares the pattern of write data coupled through the flip-flop 318 to the pattern of read data coupled through the flip-flop 330 to determine whether the memory device 240 was able to correctly store the write data. The results of the data comparison are then output from the compare circuit 340 and stored in a results memory 346, which may be a static random access memory (“SRAM”) device. The results memory 346 stores results data for one or more data patterns stored in the memory device 240, and then couples the stored results data from the results memory 346, preferably through a maintenance port 350.

[041] In operation, the BIST engine 310 adjusts the delay of the write delay line 320 to skew the write data signals from the write flip-flop 318 in relation to the write strobe signal generated from the ICLK signal. By ensuring that data can be stored in the memory device 240 during production testing despite abnormal variations in the timing between the write data signals and the write strobe signal, the manufacturer of the memory modules can be assured that the memory module will perform satisfactory in normal use. Similarly, the BIST engine 310 adjusts the delay of the read delay line 334 during production testing to vary the timing of the read data signals relative to the read data strobe. Further, the self-test module 300 can perform these operations at an abnormally high operating speed by increasing the frequency of the CLK signal generated by the clock generator 324.

[042] From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.